INFORMATICA, 1994, Vol. 5, No. 3–4, 338–350

SOLVING DIFFERENTIAL EQUATIONS BY EVENT DRIVEN TECHNIQUES FOR PARAMETER OPTIMIZATION

Henrikas KURYLA and Jonas MOCKUS

Institute of Mathematics and Informatics
2600 Vilnius, Akademijos St. 4, Lithuania

Abstract. In this paper we consider the numerical solution of the large system of nonlinear differential equations. We assume that the system simulates the semiconductor circuit. We apply the well known event driven techniques to get some approximation of solution fast. We extend those techniques by considering pairs of nodes, instead of single nodes, as usual. The "pairwise"solution is more efficient in tightly coupled circuits. The improvement of efficieny of solution is important optimizing the parameters of circuits. In many cases we need the global optimization of equation parameters. Then, the modeling speed could be the main factor of successful application.

Key words: numerical methods, differential equations, circuits, optimization.

1. Event driven techniques. In the circuit simulation we represent each of the nonlinear device in the circuit by a time-invariant network with nonlinear resistors, nonlinear capacitors, and nonlinear current sources. Using a nodal formulation (Chua and Lin, 1975) we get

$$C(x)\dot{x} + G(x)x = U(t), \tag{1}$$

where x represents the voltages of n nodes, $C(x) = (c_{ij}(x))$, $i, j = 1, ..., n$ is the matrix of nonlinear capacitors, $G(x) = (g_{ij}(x))$, $i, j = 1, ..., n$ is the matrix of nonlinear conductances and $U(t) = (u_i(t))$, $i = 1, ..., n$ represents a time-dependent input. We assume that the order of Eq. 1 is equal to the number of nodes n.

We represent nonlinear differential Eq. 1 as a sequence of linear differential equations. We do it by fixing the voltage-dependent parameters $C(x^k) = C^k$ and $G(x^k) = G^k$ at some discrete values $x = x^k$, corresponding to the moments $t = t_k$. We fix time-dependent parameters $U(t_k) = U^k$ at the moments $t = t_k$. We update those parameters after each step $k = 0, 1, 2, ...$ of approximate solution. Thus for any node i we can write

$$c_{ii}^k \frac{dx_i}{dt} + g_{ii}^k x_i - \sum_{j \in J_i} c_{ij}^k \frac{dx_j}{dt} - \sum_{j \in J_i} g_{ij}^k x_j = u_i^k, \qquad (2)$$
$$i = 1, ..., n, \quad k = 0, 1, 2, \ldots, K.$$

Here J_i is the set of nodes adjacent to node i, $c_{ij} = c_{ji}$, and $g_{ij} = g_{ji}$. Increasing the number of steps K the solution of linearization (2) may approach the solution of nonlinear equations (1).

The popular system of semiconductor circuit simulation SPICE (Nagel, 1975), while solving the equations similar to (2) uses sparse matrix techniques (Crout, 1941). Thus the results of simulation may converge to the exact solution of the system (1). However the amount of calculations using SPICE is to large to carry out the multiply simulation, which usually is needed if we wish to optimize the circuit.

The alternative system CINNAMON approximately solving (2) uses event driven techniques (Freitas, 1987; Vidigal *et al.*, 1986). The technique considers only one node at a time, keeping the remaining nodes "constant". It means that in system (2) the voltages x^k and their derivatives $\dot{x}^k$ at the adjacent nodes remain constant until the next $k + 1$ step.

The CINNAMON system defines the next moment t^{k+1} as the minimal time to get the significant change of voltage. It means that we fix the set of voltage values x^k, but not the time moments t_k, as usual. Thus the next moment t^{k+1} is the nearest moment when the voltage reach the next (higher or lower) discrete value x_i^k. We call this moment as the event. The event $k + 1$ means that we update values of $x = x^{k+1}$, $\dot{x} = \dot{x}^{k+1}$, $C = C^{k+1}$, and $G = G^{k+1}$.

We may get the next event t^{k+1} by solving linear differential Eq. 2 for each node $i = 1, ..., n$ while keeping the voltages of other nodes

constant. So we define the "critical"moment t_i^k when the nearest discrete value of the voltage x_i^{k+1} is reached. We can do it by analytical solution of (2) for $i = 1, ..., n$. Then

$$t^{k+1} = \min_{1 \leqslant i \leqslant n} t_i^{k+1}. \tag{3}$$

If there are no tightly coupled nodes then this technique usually works very well. However we get a lot of trouble, if there is at least one pair of tightly coupled nodes, what is rather usual.

The main idea of this paper is to extend the event driven techniques to the case of tightly coupled nodes. We can do it by considering separately not single nodes but pairs of nodes. We may get the analytical solution of the pairs (i, j) of linear differential Eq. 2. Thus we can define the "critical"moments t_i^k for each node i and may get the next event by expression (33).

Obviously the specific algorithm in the "pairwise"case is more complicated as compared to single node one. However it pays, if some nodes are tightly coupled. We shall describe the pairwise algorithms later, in the Section 3.

The "pairwise"event driven techniques usually work well, but need more calculations. We can reduce the unnecessary calculations if we consider not the all pairs of adjacent nodes but only tightly coupled pairs. Thus the pairwise technique turns into the single node technique of CINNAMON type, if there are no tightly coupled nodes.

2. Critical moment. We may define the critical moments $t_i^{k+1} = t_i^{k+1}(j)$ by the analytical solution of the pair (i, j) of linear differential equations (2), keeping the remaining nodes "constant". This is the main part of the pairwise algorithm, therefore we shall give the complete description. From (2) writing all constants on the right side we get

$$c_{ii}^k \frac{dx_i}{dt} + g_{ii}^k x_i - c_{ij}^k \frac{dx_j}{dt} - g_{ij}^k x_j = \sum_{l \in J_i \setminus j} c_{il}^k \frac{dx_l}{dt} + \sum_{l \in J_i \setminus j} g_{il}^k x_l + u_i^k,$$

$$c_{jj}^k \frac{dx_j}{dt} + g_{jj}^k x_j - c_{ji}^k \frac{dx_i}{dt} - g_{ji}^k x_i = \sum_{l \in J_j \setminus i} c_{jl}^k \frac{dx_l}{dt} + \sum_{l \in J_j \setminus i} g_{lj}^k x_l + u_j^k. \tag{4}$$

Denote the right parts by b_i and b_j correspondingly

$$\begin{aligned} c_{ii}^k \frac{dx_i}{dt} + g_{ii}^k x_i - c_{ij}^k \frac{dx_j}{dt} - g_{ij}^k x_j &= b_i, \\ c_{jj}^k \frac{dx_j}{dt} + g_{jj}^k x_j - c_{ji}^k \frac{dx_i}{dt} - g_{ji}^k x_i &= b_j, \end{aligned} \tag{5}$$

where

$$\begin{aligned} b_i &= \sum_{l \in J_i \setminus j} c_{il}^k \frac{dx_l}{dt} + \sum_{l \in J_i \setminus j} g_{il}^k x_l + u_i^k, \\ b_j &= \sum_{l \in J_j \setminus i} c_{jl}^k \frac{dx_l}{dt} + \sum_{l \in J_j \setminus i} g_{lj}^k x_l + u_j^k. \end{aligned} \tag{6}$$

The coefficients A_i, A_j and $\bar{A}_i, \bar{A}_j$ of partial solutions we define by the following equations

$$\begin{aligned} (c_{ii}r + g_{ii})A_i - (c_{ij}r + g_{ij})A_j &= 0, \\ -(c_{ji}r + g_{ji})A_i + (c_{jj}r + g_{jj})A_j &= 0, \\ (c_{ii}\bar{r} + g_{ii})\bar{A}_i - (c_{ij}\bar{r} + g_{ij})\bar{A}_j &= 0, \\ -(c_{ji}\bar{r} + g_{ji})\bar{A}_i + (c_{jj}\bar{r} + g_{jj})\bar{A}_j &= 0. \end{aligned} \tag{7}$$

The eigen-values r and $\bar{r}$ in (7) we define by the characteristic equation

$$\begin{vmatrix} c_{ii}r + g_{ii} & -c_{ij}r - g_{ij} \\ -c_{ji}r - g_{ji} & c_{jj}r + g_{jj} \end{vmatrix} = 0. \tag{8}$$

Solving (8) we get

$$r = -p/2 + \sqrt{p^2/4 - q + \epsilon}, \tag{9}$$

$$\bar{r} = -p/2 - \sqrt{p^2/4 - q + \epsilon}, \tag{10}$$

where

$$p = \frac{g_{ii}c_{jj} + g_{jj}c_{ii} - c_{ij}g_{ij} - c_{ji}g_{ji}}{c_{ii}c_{jj} - c_{ij}c_{ji}}, \tag{11}$$

$$q = \frac{g_{ii}g_{jj} - g_{ij}^2}{c_{ii}c_{jj} - c_{jj}^2}. \tag{12}$$

Here $\epsilon > 0$ is some small number which keeps r and $\bar{r}$ different to avoid unnecessary complications.

Substituting r and $\bar{r}$ from (9) and (10) into expression (7) we can write

$$A_j = \frac{c_{ii}r + g_{ii}}{c_{ij}r + g_{ij}}A_i = \frac{c_{ji}r + g_{ji}}{c_{jj}r + g_{jj}}A_i, \tag{13}$$

$$\bar{A}_j = \frac{c_{ii}\bar{r} + g_{ii}}{c_{ij}\bar{r} + g_{ij}}\bar{A}_i = \frac{c_{ji}\bar{r} + g_{ji}}{c_{jj}\bar{r} + g_{jj}}\bar{A}_i. \tag{14}$$

Now the partial solutions of homogeneous equations (5)

$$\begin{aligned} x_i &= A_i e^{rt} = D_i e^{rt}, \\ x_j &= A_j e^{rt} = hD_i e^{rt}, \end{aligned} \tag{15}$$

and

$$\begin{aligned} x_i &= \bar{A}_i e^{\bar{r}t} = D_j e^{\bar{r}t}, \\ x_j &= \bar{A}_j e^{\bar{r}t} = \bar{h}D_i e^{\bar{r}t}. \end{aligned} \tag{16}$$

Here D_i and D_j are the integration constants and

$$h = \frac{c_{ii}r + g_{ii}}{c_{ij}r + g_{ij}} = \frac{c_{ji}r + g_{ji}}{c_{jj}r + g_{jj}},$$

$$\bar{h} = \frac{c_{ii}\bar{r} + g_{ii}}{c_{ij}\bar{r} + g_{ij}} = \frac{c_{ji}\bar{r} + g_{ji}}{c_{jj}\bar{r} + g_{jj}}.$$

Thus the general solution of the homogeneous equations (5)

$$\begin{aligned} x_i &= D_i e^{rt} + D_j e^{\bar{r}t}, \\ x_j &= hD_i e^{rt} + \bar{h} D_j e^{\bar{r}t}. \end{aligned} \tag{17}$$

We can get the partial solution of nonhomogeneous equations (5) by substituting the general solution x_i and x_j (17) of corresponding homogenious part of (5) into complete nonhomogeneous equations (5), and assuming that D_i and D_j are not constants but depend on time $D_i = D_i(t)$, $D_j = D_j(t)$. Thus

$$\begin{aligned} c_{ii}\dot{D}_i e^{rt} + c_{ii}\dot{D}_j e^{\bar{r}t} - c_{ij}h\dot{D}_i e^{rt} - c_{ij}\bar{h}\dot{D}_j e^{\bar{r}t} &= b_i, \\ -c_{ji}\dot{D}_i e^{rt} - c_{ji}\dot{D}_j e^{\bar{r}t} + c_{jj}h\dot{D}_i e^{rt} + c_{jj}\bar{h}\dot{D}_j e^{\bar{r}t} &= b_j, \end{aligned} \tag{18}$$

or

$$\begin{aligned} (c_{ii} - c_{ij}h)\dot{D}_i e^{rt} + (c_{ii} - c_{ij}\bar{h})\dot{D}_j e^{\bar{r}t} &= b_i, \\ (-c_{ji} + c_{jj}h)\dot{D}_i e^{rt} + (-c_{ji} + c_{jj}\bar{h})\dot{D}_j e^{rt} &= b_j. \end{aligned} \tag{19}$$

Here $\dot{D}_i = dD_i(t)/dt$ and $\dot{D}_j = dD_j(t)/dt$.
Denote

$$\begin{aligned} a_i &= c_{ii} - c_{ij}h, \qquad \bar{a}_i = c_{ii} - c_{ij}\bar{h}, \\ a_j &= -c_{ji} + c_{jj}h, \quad \bar{a}_j = -c_{ji} + c_{jj}\bar{h}, \end{aligned} \tag{20}$$

and

$$\begin{aligned} v_i &= a_i e^{rt}, \qquad \bar{v}_i = \bar{a}_i e^{\bar{r}t}, \\ v_j &= a_j e^{rt}, \qquad \bar{v}_j = \bar{a}_j e^{\bar{r}t}. \end{aligned} \tag{21}$$

Then from (19)

$$\begin{aligned} \dot{D}_i v_i + \dot{D}_j \bar{v}_i &= b_i, \\ \dot{D}_i v_j + \dot{D}_j \bar{v}_j &= b_j. \end{aligned} \tag{22}$$

From (22)

$$\begin{aligned} \dot{D}_i &= \frac{b_i\bar{v}_j - b_j\bar{v}_i}{v_i\bar{v}_j - v_j\bar{v}_i}, \\ \dot{D}_j &= \frac{b_iv_j - b_jv_i}{\bar{v}_iv_j - \bar{v}_jv_i}. \end{aligned} \tag{23}$$

From (23) and (21)

$$\begin{aligned} \dot{D}_i &= \frac{b_i\bar{a}_j - b_j\bar{a}_i}{a_i\bar{a}_j - a_j\bar{a}_i}\, e^{-rt}, \\ \dot{D}_j &= \frac{b_ia_j - b_ja_i}{\bar{a}_ia_j - \bar{a}_ja_i}\, e^{-\bar{r}t}. \end{aligned} \tag{24}$$

Denote

$$\begin{aligned} H_i &= \frac{b_i\bar{a}_j - b_j\bar{a}_i}{a_i\bar{a}_j - a_j\bar{a}_i}, \\ H_j &= \frac{b_ia_j - b_ja_i}{\bar{a}_ia_j - \bar{a}_ja_i}. \end{aligned} \tag{25}$$

Then

$$\begin{aligned} \dot{D}_i &= H_ie^{-rt}, \\ \dot{D}_j &= H_je^{-\bar{r}t}. \end{aligned} \tag{26}$$

After integration of (26)

$$\begin{aligned} D_i &= \begin{cases} -1/rH_ie^{-rt} + D_{i0}, & \text{if } r \neq 0, \\ H_it + D_{i0}, & \text{if } r = 0; \end{cases} \\ D_j &= \begin{cases} -1/\bar{r}H_je^{-\bar{r}t} + D_{j0}, & \text{if } \bar{r} \neq 0, \\ H_jt + D_{j0}, & \text{if } r = 0. \end{cases} \end{aligned} \tag{27}$$

From (17) and (27)

$$\begin{aligned} x_i &= -(H_i/r + H_j/\bar{r}) + D_{i0}e^{rt} + D_{j0}e^{\bar{r}t}, \\ x_j &= -(hH_i/r + \bar{h}H_j/\bar{r}) + D_{i0}e^{rt} + D_{j0}e^{\bar{r}t}. \end{aligned} \tag{28}$$

We are looking for the partial solution, therefore we may set the integration constants to zero $D_{i0} = D_{j0} = 0$. Then the partial solution of nonhomogeneous equations (5)

$$\begin{aligned} x_i &= -(H_i/r + H_j/\bar{r}), \\ x_j &= -(hH_i/r + \bar{h}H_j/\bar{r}). \end{aligned} \tag{29}$$

Expressing the general solution of nonhomogeneous equation (5) as the sum of the general solution (17) of homogeneous equation (5) and the partial solution (29) of nonhomogeneous equation (5) we get

$$\begin{aligned} x_i &= -(H_i/r + H_j/\bar{r}) + D_i e^{rt} + D_j e^{\bar{r}t}, \\ x_j &= -(hH_i/r + \bar{h}H_j/\bar{r}) + hD_i e^{rt} + \bar{h}D_j e^{\bar{r}t}. \end{aligned} \tag{30}$$

Here D_i and D_j are constants. We shall define those constants by the initial values of x. Denote the values of x_i and x_j at the zero moment $t = 0$ as x_{i0} and x_{j0} correspondingly.

Denote

$$\begin{aligned} H &= -(H_i/r + H_j/\bar{r}), \\ \bar{H} &= -(hH_i/r + \bar{h}H_j/\bar{r}). \end{aligned}$$

Then the initial values

$$\begin{aligned} x_{i0} &= H + D_i + D_j, \\ x_{j0} &= \bar{H} + hD_i + \bar{h}D_j. \end{aligned} \tag{31}$$

We can define constants D_i and D_j by solving equations (31)

$$\begin{aligned} D_i &= \frac{(x_{j0} - \bar{H}) - (x_{i0} - H)\bar{h}}{h - \bar{h}}, \\ D_j &= \frac{(x_{i0} - H)h - (x_{j0} - \bar{H})}{h - \bar{h}}. \end{aligned} \tag{32}$$

3. The pairwise event driven techniques. Now we specify the order in which we shall process nodes. We will make as few changes as

possible as compared to the well known single event driven technique CINNAMON. First we recall the CINNAMON algorithm:

Step 0. Set the starting time $t = t^0$. Linearize the equations at the starting voltage level $x_i = x_i^0$.

Step 1. Solve equations (2) for each node i. Determine the time t_i^1 when the node i will mark the next event.[1]

Step 2. Choose the node i^{k+1} with the smallest time $t^{k+1} = \min_i t_i^{k+1}$ and set the corresponding voltage to the new level $x_{i^{k+1}} = x_{i^{k+1}}^{k+1}$.

Step 3. Linearize the equations at the new voltage level $x_{i^{k+1}}^{k+1}$ for all the nodes i adjacent to the node i^{k+1}.

Step 4. Set the time for this group of nodes to t^{k+1} and return to *Step* 2.

The main idea of CINNAMON is that the perturbation induced in some node causes perturbations in the adjacent nodes and so propagates through the circuit. We would like to take advantage of this wave-like behaviour of algorithm in the pairwise technique, too. We call the pairwise technique as CINNAMON2.

Suppose that the event happened in the node i. In the single node technique we consider all the voltage values and their derivatives of the adjacent nodes as constant. We solve a single equation for each single node i and so define the next event time t_i^{k+1}.

Using pairwise technique we solve the systems of two equations for the pairs of the node i and all the adjacent nodes j. Denote by t_i^{k+1} the time when the voltage of the event node i reaches the next discrete level.

The event time t_i^{k+1} for the fixed event node i depends on the "partner"node j, too. It means that $t_i^{k+1} = t_i^{k+1}(j)$. We choose the smallest $t_i^{k+1}(j)$ as the next event time t^{k+1}.

$$t^{k+1} = \min_j t_i^{k+1}(j). \tag{33}$$

Now we describe the pair-wise algorithm:

[1] The moment when the voltage x_i will reach the next discrete level.

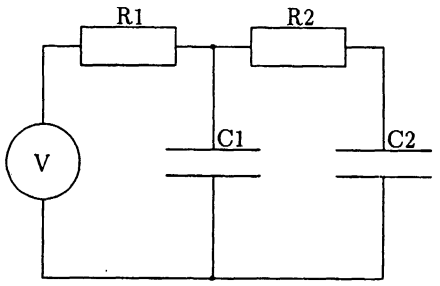

Fig. 1. Two RC circuit.

Step 0. Set the starting time $t = t^0$. Linearize the equations at the starting voltage level $x_i = x_i^0$.

Step 1. Solve equations (2) for each node i. Determine the time t_i^1 when the node i will mark the next event.

Step 2. Choose the node i^{k+1} with the smallest time $t^{k+1} = \min_i t_i^{k+1}$ and set the corresponding voltage to the new level $x_{i^{k+1}} = x_{i^{k+1}}^{k+1}$.

Step 3. Linearize the equations at the new voltage level $x_{i^{k+1}}^{k+1}$ for all the nodes i adjacent to the node i^{k+1}.

Step 4. Define the times $t_i^{k+1}(j)$ corresponding to each adjacent node j by solving the pairs (i, j) of equations (2). Determine the times t_j^{k+1} when the nodes j will mark the next events.

Step 5. Choose for time t^{k+1} the smallest of $t_i^{k+1}(j)$.

Step 6. Set the time t^{k+1} for the node i and all the adjacent nodes j adjacent to node i. Return to *Step* 2.

4. Computing results. We consider the three techniques:

i. The single node CINNAMON (Vidigal *et al.*, 1986);

ii. The pairwise CINNAMON2;

iii. The well known circuit simulation softwarem SPICE directly solving system (1) and using the sparse matrix techniques (Crout, 1941; White and Sangiovanni–Vincentelli, 1986).

Example 1. We consider the circuit (Fig. 1) of two RC elements, where C1 = C2 = 1 and R1 = 500, R2 = 1, V = 5. Fig. 2 shows how the

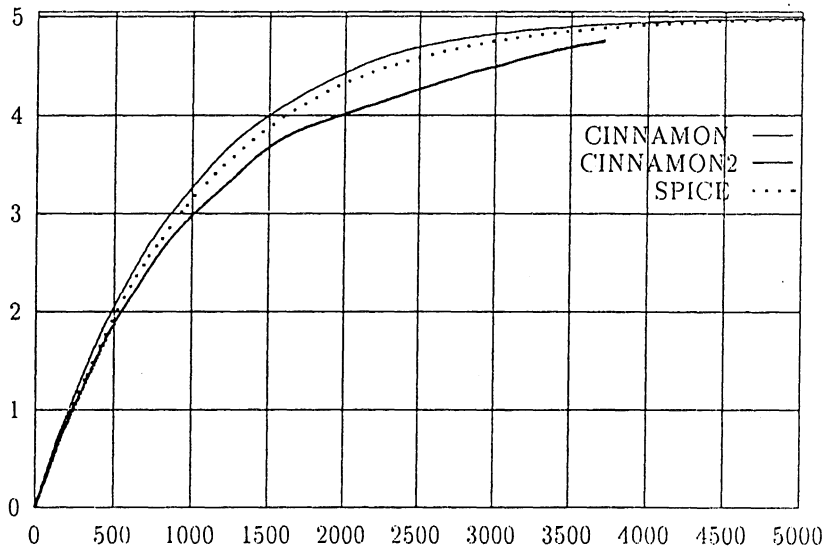

Fig. 2. Comparing voltages by three techniques. Two RC circuit. Weak coupling.

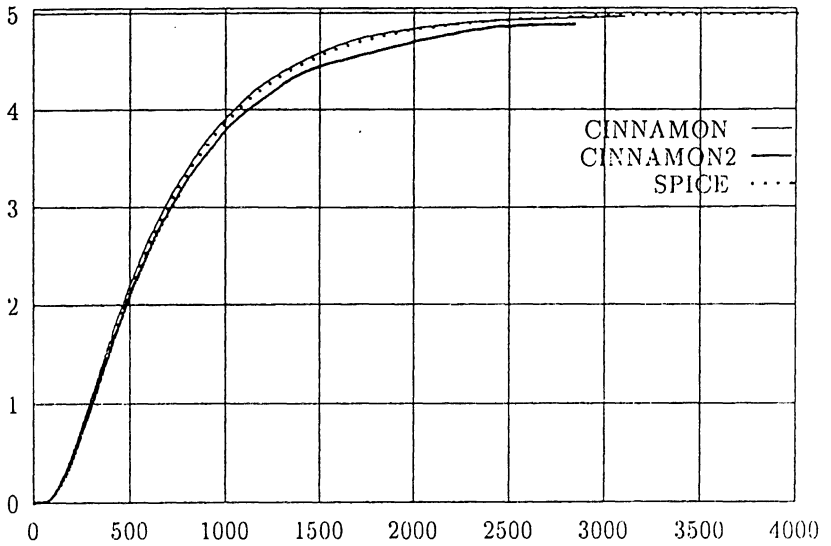

Fig. 3. Comparing voltages by three techniques. Ten RC circuit. Weak coupling.

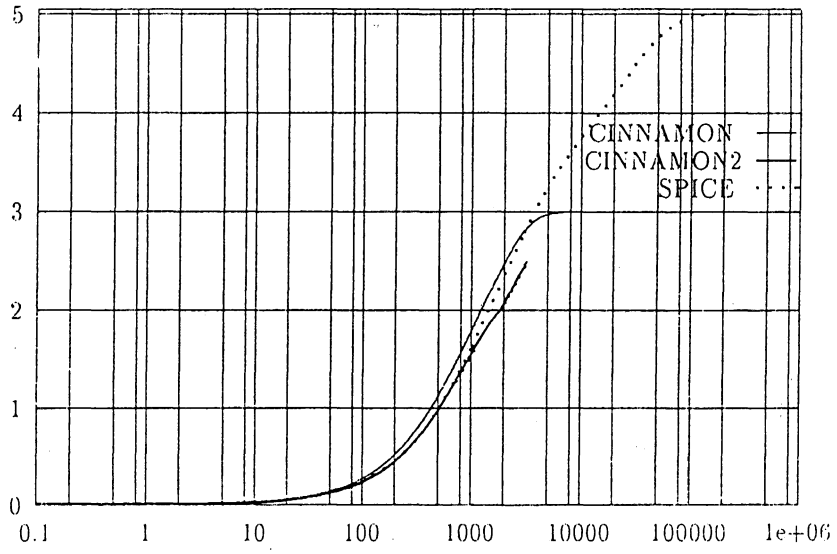

Fig. 4. Comparing voltages by three techniques. Ten RC circuit. Strong coupling.

voltage of node 3 depends on time. Using CINNAMON algorithm we see some voltage delay. The results of both CINNAMON2 and SPICE algorithms are close enough.

Example 2. In this example we consider the circuit of ten RC elements, where C1 = C2 = ... = C10 = 1 , R1 = R2 = ... = R8 = 10, R9 = 100, R10 = 1, and V=5.

Fig. 3 shows how the voltage of node 11 depends on time. As in the previous example CINNAMON voltage delays, while the both other algorithms show similar results.

Example 3. We consider the same circuit as in example 2, but we change the resistances: R1 = R3 = R5 = R7 = R9 = 1000, R2 = R4 = R6 = R8 = R10 = 1, V = 5. Fig. 4 shows the voltage of node 11. We see that the pairwise CINNAMON2 and the single node CINNAMON don't reach the stable voltage level. The example shows that with strong coupling betveen serial nodes the results the CINNAMON2 algorithm are between SPICE and CINNAMON algorithms, but closer to SPICE.

REFERENCES

Chua, L.O., and P.M.Lin (1975). *Computer-Aided Analysis of Electronic Circuits.* Prentice–Hall.

Crout, P.D. (1941). A short method for evaluating determinants and solving systems of linear equations with real and complex coefficients. *AIEE Transactions,* **69**, 1235–1241.

Freitas, P. (1987). Solving differential equations using event driven techniques. *Report Nr. ESPRIT1058/INESC/6.87/D8728,* Lisboa, Portugal.

Mockus, J. (1989). *Bayesian Approach to Global Optimization.* Kluwer Academic Publishers, Dordrecht–Boston–London.

Nagel, L. (1975). SPICE2: A computer program to simulate semiconductor cirquits. *Technical Report, UCB ERL–M250,* University of California, Berkeley.

Vidigal, L.M., S.R.Nassif and S.W.Director (1986). CINNAMON: Coupled Integration and Nodal Analysis of MOS Networks. *23-th ASM/IEEE Design Automation Conference.*

White, J., and A.Sangiovanni–Vincentelli (1986). *Relaxation Techniques for the Simulation of VLSI Circuits.* Kluwer Academic Publishers, Dordrecht–Boston–London.

Received May 1994

H. Kuryla was born in 1969. He graduated the Faculty of Informatics of the Kaunas Technological University in 1992. He is a junior research fellow at the Optimal Decisions Theory Department of the Institute of Mathematics and Informatics. His research interests are applications of global optimization.

J. Mockus graduated Kaunas Technological University, Lithuania, in 1952. He got his Doctor habilitus degree in the Institute of Computers and Automation, Latvia, in 1967. He is a head of Optimal Decision Theory Department, Institute of Mathematics and Informatics, Vilnius, Lithuania and professor of Kaunas Technological University.